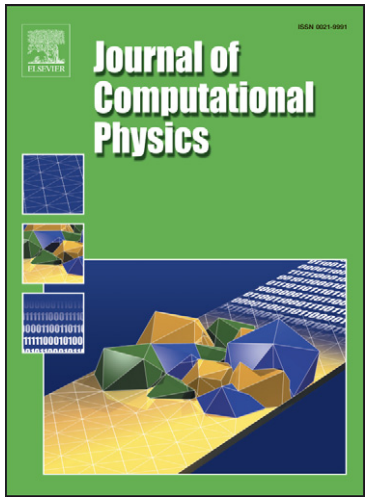

Implementation of Unsteady Sampling Procedures for the Parallel Direct Simulation Monte Carlo Method

H.M. Cave, K.-C. Tseng, J.-S. Wu, M.C. Jermy, J.-C. Huang, S.P. Krumdieck

PII: S0021-9991(08)00154-X
DOI: 10.1016/j.jcp.2008.03.015
Reference: YJCPH 2054

To appear in: *Journal of Computational Physics*

Received Date: 22 June 2007
Revised Date: 31 December 2007
Accepted Date: 7 March 2008

Please cite this article as: H.M. Cave, K.-C. Tseng, J.-S. Wu, M.C. Jermy, J.-C. Huang, S.P. Krumdieck, Implementation of Unsteady Sampling Procedures for the Parallel Direct Simulation Monte Carlo Method, *Journal of Computational Physics* (2008), doi: 10.1016/j.jcp.2008.03.015

Implementation of Unsteady Sampling Procedures for the Parallel Direct Simulation Monte Carlo Method

H.M. Cave[1], K.-C. Tseng[2], J.-S. Wu[3], M.C. Jermy[1], J.-C. Huang[4] and S.P. Krumdieck[1]*

[1] *Department of Mechanical Engineering, University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand.*

[2] *National Space Organisation, 8F, 9 Zhan-Ye 1st Road, Hsinchu Science Park, Hsinchu, Taiwan.*

[3] *Department of Mechanical Engineering, National Chiao-Tung University, 1001 Ta-Hsueh Road, Hsinchu 30050, Taiwan.*

[4] *Department of Computer Science and Information Engineering, Nanhua University, 32 Chung Keng Li, Dalin, Chia-Yi 62248, Taiwan.*

[*] *Corresponding author:* *Phone: +886-3-573-1693*

Fax: +886-3-611-0023

Email: chongsin@faculty.nctu.edu.tw

Submitted to: Journal of Computational Physics

Manuscript Type: Full-length article

Date Submitted: June 2007

Date Revised: December 31, 2007

Abstract

An unsteady sampling routine for a general parallel Direct Simulation Monte Carlo method called PDSC is introduced, allowing the simulation of time-dependent flow problems in the near continuum range. A post-processing procedure called DSMC Rapid Ensemble Averaging Method (DREAM) is developed to improve the statistical scatter in the results while minimising both memory and simulation time. This method builds an ensemble average of repeated runs over small number of sampling intervals prior to the sampling point of interest by restarting the flow using either a Maxwellian distribution based on macroscopic properties for near equilibrium flows (DREAM-I) or output instantaneous particle data obtained by the original unsteady sampling of PDSC for strongly non-equilibrium flows (DREAM-II). The method is validated by simulating shock tube flow and the development of simple Couette flow. Unsteady PDSC is found to accurately predict the flow field in both cases with significantly reduced run-times over single processor code and DREAM greatly reduces the statistical scatter in the results while maintaining accurate particle velocity distributions. Simulations are then conducted of two applications involving the interaction of shocks over wedges. The results of these simulations are compared to experimental data and simulations from the literature where there these are available. In general it was found that ten ensembled runs of DREAM processing could reduce the statistical uncertainty in the raw PDSC data by 2.5-3.3 times, based on the limited number of cases in the present study.

Keywords: DSMC, unsteady, shock tube, shock impingement, Couette flow, PDSC, DREAM

Mathematical Review Index Classification: 65C99, 65Y05, 76P05

1. Introduction

The Direct Simulation Monte Carlo (DSMC) method has become a widely used computational tool for the simulation of rarefied gas flows where effects at the molecular scale become significant [1]. Applications include modelling of hypersonic flows [2], satellite thrusters at high altitude [3], micro electro mechanical system (MEMS) devices [4] and chemical vapour deposition (CVD) [5], among others.

In these high Knudsen number flows, the continuum assumption can break down meaning that the Navier-Stokes equations fail to correctly model the flow. The Boltzmann equation, which is appropriate for modelling rarefied flow, is extremely difficult to solve numerically due to its high dimensionality and the complexity of the collision term. Simplification of the collision term by the Bhatnagar-Gross-Krook (BGK) equation [6] has spawned a number of numerical solution techniques, such as Huang's model Boltzmann equation (MBE) solver [7], however these techniques are still in their infancy. DSMC offers an efficient particle based approach in which the movement and collisional behavior of a large number of representative particles within the flow field are decoupled over a time step which is a small fraction of the local mean collision time. This method has been shown mathematically to effectively provide a solution to the Boltzmann equation as the number of simulated particles becomes large [8]. A number of scalar (single-processor) DSMC codes have been developed, most notably by Bird, which utilise sophisticated features such as nearest-neighbour collisions, adaptive cell structures and variable time-step (VTS) schemes to improve the accuracy and speed of the simulations, and additionally allow the incorporation of complex effects such as non-equilibrium gas-phase chemistry [1,9].

A major drawback of the DSMC scheme is that the computational expense of the technique becomes increasingly expensive as the flow density increases. Consequently, modelling of many practical flows which are in the near-continuum regime are prohibitively computationally expensive when single computer processors are employed. The development of parallel computer processing, whereby the computational load is spread over a number of machines, represents an opportunity to simulate near-continuum flows with acceptable run-times. Additionally, the DSMC technique is ideally suited to parallelisation since the movement of each particle is independent of all others with the only local coupling required during the collision step.

In the past two decades, a number of parallel-DSMC schemes have been implemented and reported in the literature [10-13]. These schemes utilized either structured or unstructured meshes and mostly static domain decomposition. Here message passing is used to transfer molecules between processors and for simulation synchronization, however the computational speed-up due to parallelisation is limited by load imbalancing and the cost of communication between the processors. These limitations necessitate sizing the problem carefully to the number of processors. Several recent implementations of parallel-DSMC include those by the groups led by Boyd [14], Ivanov [15], LeBeau [16] and Wu [2,17-19].

Boyd's code, named MONACO, utilizes unstructured grids so that objects with complex geometry can be handled relatively readily, and static domain decomposition for the distribution of computational load. The method has been used to model flow around a planetary probe using 100 million particles and 400 IBM-SP2 processors. Ivanov's code, named SMILE, arranges the cells into "clusters" which are in turn divided among the processors using scalable dynamic domain decomposition. The code

employs a simple but effective method of indexing particles to the grid, which reduces the computational time, and determines the direction and amount of workload transfer using the concept of heat diffusion. LeBeau has developed the DSMC Analysis Code (DAC) which uses a two-level embedded Cartesian grid, which is uncoupled from the surface geometry, to discretize the computational domain. The code has been used to study the flow over a sphere using 128 processors with 90% parallel efficiency.

The Parallel DSMC Code (PDSC) developed by Wu's group has been successfully used to model a number of flows including flow through a drag pump [16], hypersonic flow past a cylinder [18] and under-expanded jet flow [19]. PDSC will be discussed in greater detail in section 2.2.

Unsteady rarefied flows, in which the flow structure changes significantly with time, are interesting flow problems with a number of applications such as the development of under-expanded jets from sonic nozzles during the start up of rocket nozzles and during the injection and initial pump-down phases of the Pulsed Pressure Chemical Vapor Deposition (PP-CVD) process [20]. Unsteady DSMC simulations have been greatly neglected in the literature, primarily because sampling over a small time interval requires either a very large number of simulated molecules or the average of a large number of separate simulations (usually termed "ensemble-averaging"). The associated high computational expense and large memory requirements mean that investigations in the literature tend to be restricted to one-dimensional problems, such as shock tube flow [21] or the shock waves generated by moving pistons [1]. Two dimensional unsteady problems have been attempted, and one method of decreasing the statistical scatter of the results is to using statistical smoothing procedures [22]. Bird's two-dimensional axisymmetric code DS2V [8] incorporates unsteady sampling techniques in which a

number of time intervals close to the sampling point are averaged (usually termed "time-averaging"), however this single processor code is unable to model processes such as PP-CVD with acceptable run-times [23]. The increased computational capacities of parallel-DSMC techniques have the potential to enable the simulation of time-dependent flow problems at the near-continuum regime.

This paper begins with a brief description of the DSMC method and the parallel DSMC code (PDSC). The development and implementation of an unsteady sampling method for PDSC, along with a post-processing methodology, is then outlined and simulations of a shock tube and the development of Couette flow are then carried out as validation studies. Results of simulations for a number of applications are then presented, including the impingement of a moving shock on several wedge configurations. These simulations are compared to the results of other studies in the literature, and to experimental data where it is available.

2. Numerical Method

2.1. The Direct Simulation Monte Carlo (DSMC) Method

The DSMC method is a particle-based method for the simulation of gas flows which was developed by Bird during the 1960s. The details of the procedures and the consequences of the computational approximations are outlined in detail in the monograph by Bird [1], so only a brief outline of the method is presented here.

In DSMC, the gas is represented at the microscopic level by simulated particles which each represent a much larger number of real particles. The physics of the gas flow are modelled through the motion of the particles and the collisions between them, however these two steps are decoupled over a time step which is a small fraction of the

mean collision time. Mass, momentum and energy are conserved at the particle level whereas physical events such as intermolecular collisions are handled probabilistically using phenomenological models. These models are designed to reproduce real fluid behavior when the flow is examined at the macroscopic level. These models vary in their sophistication, however the models used in most applications include the variable hard sphere (VHS) [24] and the variable soft sphere (VSS) [25] models. The computational domain itself is divided into either a structured or unstructured grid of cells which are then used to select particles for collisions on a probabilistic basis and also are used for sampling the macroscopic flow properties. In practice, often the sampling cells are further divided into smaller collision cells to ensure intermolecular interactions occur between closely spaced molecules [1].

In general, the DSMC procedure involves 1) moving the particles ballistically over a small time step and applying boundary conditions to particles which collide with boundaries, 2) indexing the particles within the grid of collision cells, 3) selecting particles from within the cells on a probabilistic basis and applying the collision routines to these and 4) sampling the macroscopic flow properties from the collision cells.

The DSMC method relies heavily on pseudo-random number generators for simulating the statistical nature of the underlying process. Because data variables, such as the velocity data for an individual particle, are randomly accessed from the computer's memory, it is very difficult to vectorize the DSMC code, however because particle movement and collision events are treated independently and occur locally, the code is highly suitable for parallelization.

2.2. Parallel Implementation of DSMC

The DSMC algorithm is readily parallelized through decomposition of the physical domain into groups of cells which are then distributed among the parallel processors. Each processor executes the DSMC algorithm in serial for all particles and cells in its own domain. Parallel communication between processors is required when particles cross the domain boundaries requiring particles to be transferred between processors. To achieve high parallel efficiency it is necessary to minimize the communication between processors while maintaining a balance between the computational load on each processor. In the present study, we have adapted the previously developed Parallel DSMC Code (PDSC) which has been described in detail in the papers by Wu *et al.* [16-19] and will only be outlined briefly here.

The DSMC algorithm is implemented on a two-dimensional, axisymmetric or three-dimensional unstructured mesh using a particle ray-tracing technique, which takes advantage of the cell connectivity information provided by the mesh data and is able to handle complex boundary geometry. PDSC utilizes the multi-level graph partitioning tool ParMETIS to decompose the computational domain and distribute the cells amongst the processors. A stop-at-rise (SAR) algorithm is used to determine when to dynamically repartition and re-distribute the computational load between processors based on the value of a degradation function which compares the computational cost of repartition to the idle time for each processor. The transfer of particle data between the processors only occurs when particles strike the inter-processor boundaries and after all other particles on each processor have been moved, thus minimizing communication between processors and maximizing the parallel speed-up. During calculation, the mesh can be iteratively refined using the h-refinement technique whereby local grid

points are added to improve the cell distribution according to the solution based on some adaptation criteria (for example, flow field density or local Knudsen number). Cell quality control is used to maintain the integrity of the mesh during this process.

Other special features include pressure boundary treatment, a spatial variable time-step scheme, the implementation of a conservative weighting scheme to efficiently deal with gas flows with trace species [26] and the gas phase chemistry for simulating chemical reactions in hypersonic air flows [27]. These features have been developed to enhance the computational efficiency, flexibility and utility of PDSC. The implementation of a transient sub-cell module, allowing much higher density flows to be modelled with negligible computational expense, will be discussed in a future paper along with other improvements to the code currently under development.

2.3. Unsteady Sampling Method

The PDSC code discussed in section 2.2 has been specifically designed for simulating steady flows, so some modification is required for unsteady sampling. Two methods for unsteady sampling exist, the differences between which are illustrated in figure 1. The first, termed “ensemble-averaging”, is shown in figure 1a and requires multiple simulation runs. During each run, the flow field is sampled at the appropriate sampling times and the samples from each run are averaged over the runs to provide the flow field output. The results are very accurate, however the method is very computationally expensive because a large number of sequential runs are required to reduce the statistical scatter to an acceptably low level and a large amount of memory is required to record the sampling data for each simulation.

The second method, termed "time-averaging", is shown in figure 1b and averages a number of time steps over an interval centred on the sampling time. This method only requires one simulation run, however it suffers a potential disadvantage in that the results will be "smeared" over the time during which samples are taken. This occurs because the particles will propagate downstream over the sampling interval, resulting in a broadening of any areas with large macroscopic gradients, such as shocks, if the sampling interval is too long. Hence the sample time must be sufficiently short to minimize time "smearing" and yet long enough to obtain a good statistical sample. Similar methods of time averaging have been used previously by Auld to model shock tube flow [21] and in Bird's DS2V code.

In PDSC, the method of time-averaging was implemented. Here a technique called the temporal variable time step (TVTS) method was used to reduce the simulation time by increasing the time step between sampling. The code has an option for the user to choose specific output flow times or for output at regular intervals. Figure 2 shows the flow chart of the parallel DSMC method for *np* processors with the unsteady sampling procedures implemented. Here *M* is the output matrix for sampling interval *M*. Most parts of the procedure are the same as the steady simulation except the sampling data must be reset after completing each simulation interval.

2.4. DSMC Rapid Ensemble Averaging Method (DREAM)

Despite the efficient implementation of unsteady sampling procedures on parallel computers, simulating denser flows in reasonable computational times requires somewhat of a compromise on the statistical scatter in the results. This is because reducing the statistical scatter significantly in time-averaged data necessitates a very

large number of simulation particles with consequent large computational times. Other researchers have attempted to use data smoothing to prepare their results for presentation [22], however ultimately this removes data which may have physical significance.

The approach we have developed is outlined in figure 3. Here we select a raw data set *X*-*n* produced by PDSC *n* sampling intervals prior to the sampling interval of interest *X*. For near-continuum flow (for example, shocks of less than approximately Mach 2) new particle data is generated from the macroscopic properties in data set *X*-*n* by assuming a Maxwellian distribution of velocities based on the three components of temperature T_x, T_y and T_z (this version of the code is called DREAM-I). The DREAM-I method has the advantage of easier implementations and requires no additional input/output during the initial PDSC run, however for strongly non-equilibrium flows it may be unable to recover the correct particle velocity distribution at the sampling point. Thus, for strongly non equilibrium flows (i.e. higher Mach number shocks), the particle data is regenerated from the instantaneous particle data which can be outputted by PDSC in the original run, thus preserving the true phase-space data (DREAM-II). The standard PDSC algorithm is then used to simulate forward in time until the sampling period of interest *X* is reached. The time steps close to the sampling point are time-averaged in the same way as in PDSC and this process is repeated a number of times, thus building up a combination of ensemble- and time-averaged data without having to simulate from zero flow time for each run. This process decreases the statistical scatter in the results by adding to the number of particles in the sample, rather than by some artificial smoothing process. Because only a short period of the flow is processed in this way, the scheme has significant memory and computational advantages over

ensemble-averaging and results in a greater number of sampling particles than the time-averaging scheme.

For DREAM to be accurate there must be a suitably large number of time steps between the particle regeneration and the sampling data sets so that 1) the velocity distribution can "relax" sufficiently quickly towards the true distribution in any non-Maxwellian regions and 2) so that the macroscopic properties at the regeneration data set will not overly constrain the data at the sampling time step (i.e. to ensure that particles can move out of their original cells before being re-sampled).

For stronger non-equilibrium flows, DREAM-II reloads the original PDSC phase-space data. This allows the particle data to be regenerated using the true velocity distribution in non-equilibrium regions. However DREAM-II does have a disadvantage in terms of storing the output particle data sets during the original PDSC run. Fortunately, due to the low cost of hard-drive storage, the process is not overly expensive even though particle data must be outputted from PDSC during every sampling period if the regions for post-processing by DREAM are not known *a priori*.

3. Code validation

3.1. Shock tube flow

3.1.1. Validation of unsteady sampling procedures

As a validation of the unsteady sampling techniques employed in the PDSC code, we have used the test problem of shock tube flow. Figure 4 shows the typical flow structure in a shock tube, in which a shock wave is created by bursting a diaphragm between a high-pressure and low-pressure gas. These devices are used to investigate a

wide variety of physical phenomena including shock structures and high temperature gas reactions.

The Riemann continuum solution for a shock tube allows the properties of the flow structure, including the shock propagation velocity W, the contact surface velocity u_P along with the pressure, temperature and density, to be determined at any given time. Further details on the derivation of these equations, along with expressions for the variation of properties in the expansion fan, can be found in the monograph by Anderson [28]. These expressions are summarised for completeness below:

$$W = a_1\sqrt{\frac{\gamma+1}{2\gamma}\left(\frac{p_2}{p_1}-1\right)+1} \tag{1}$$

$$u_P = \frac{a_1}{\gamma}\left(\frac{p_2}{p_1}-1\right)\sqrt{\frac{\frac{2\gamma}{\gamma+1}}{\frac{p_2}{p_1}+\frac{\gamma-1}{\gamma+1}}} \tag{2}$$

$$\frac{p_4}{p_1} = \frac{p_2}{p_1}\left[1-\frac{(\gamma-1)\left(a_1/a_4\right)\left(p_2/p_1-1\right)}{\sqrt{2\gamma\left\{2\gamma+(\gamma+1)\left(p_2/p_1-1\right)\right\}}}\right]^{-2\gamma/(\gamma-1)} \tag{3}$$

$$\frac{p_3}{p_4} = \left(\frac{p_3}{p_1}\right)\left(\frac{p_1}{p_4}\right) = \left(\frac{p_2}{p_1}\right)\left(\frac{p_1}{p_4}\right) = \left(\frac{\rho_3}{\rho_4}\right)^{\gamma} = \left(\frac{T_3}{T_4}\right)^{\gamma/(\gamma-1)} \tag{4}$$

where a_x, p_x, ρ_x and T_x are respectively the speed of sound, pressure, density and temperature in the region x and γ is the ratio of specific heats.

To validate the PDSC code, simulations were conducted on a quasi one-dimensional shock tube of length 0.1m and width 0.0125m with argon as the working gas (the VHS molecular model was used). The upper and lower walls were implemented as specular walls to preserve the one-dimensional nature of the flow, while the end walls were simulated as diffusive walls at 300K. The initial conditions in the high pressure and

low pressure ends of the shock tube are p_4 = 100Pa and p_1 = 10Pa respectively, however the temperatures at both ends of the tube are the same such that $T_4 = T_1$ = 300K. This results in a shock Mach number of 1.55. This low shock number is a challenging simulation for a DSMC method, since the thermal and macroscopic velocities will be of a similar order of magnitude. This means that insufficient sampling will result in high statistical scatter in macroscopic properties.

The solution was first computed using Bird's DS2V code (version 3.7.03) using its standard settings and then a run was conducted using PDSC using a similar number of cells and particles. The solution was also compared to the results generated by a one-dimensional ensemble-averaging code implemented in MATLAB™ (called HDSMC), using the same flow conditions but with a cell size set to approximately one-third of the average mean free path, as recommended by Bird [1]. The DS2V and HDSMC runs were conducted on a single processor Pentium IV 3.2GHz (hyper-threading enabled), while the PDSC run was conducted on a PC cluster system of ten Athlon XP2100s. Each run is detailed in table 1. The conditions for the PDSC run were selected to attempt to preserve the simulation conditions from the DS2V run.

Figure 5 shows the pressure, density and temperature profiles at 27.45μs as generated by the three methods, along with the Riemann continuum solution. All three methods capture the flow profile accurately with the positions and general structure of each flow feature comparing well with the continuum solution. As would be expected the sharp continuum solution is not followed exactly since it does not include viscous effects. The PDSC results show similar scatter to the DS2V solution, however the PDSC solution does not exhibit the "spikes" in the temperature profile predicted by DS2V, which can be seen more clearly in figure 6. Additionally, parallelisation means

PDSC runs approximately ten times faster than DS2V, and has considerably reduced statistical scatter over the HDSMC ensemble averaging solution.

Figure 6 shows the evolution of the shock structure predicted by PDSC and DS2V. Both methods predict equivalent flow profiles, with equivalent amounts of statistical scatter. At 27.5μs and 76.9μs, the incident shock can be seen advancing toward the right hand end of the shock tube where it impacts with the wall at approximately 99.8μs. After the shock has been reflected from the wall there is a sharp rise in density as the reflected shock begins to travel to the left. This reflected shock interacts with the advancing contact surface, creating the peak in density visible at 175.7μs. At the left hand end of the tube, the expansion fan also reflects from the solid wall, creating the reduction in density visible at 175.7μs. In figure 6, PDSC also does not exhibit the spike in density predicted by DS2V, as mentioned earlier.

The sensitivity of the PDSC flow field solution to changes in cell size and average number of particles per cell was also investigated. Figure 7 shows the effect of the number of particles in each collision cell on the temperature profile at 27.45μs (which is statistically the most sensitive measurement) for a range of different numbers of particles per cell. The continuum solution is not shown for clarity. It can be seen from these results that the flow profile is insensitive to the number of simulated particles, however, as would be expected, the statistical uncertainty increases as the number of particles is reduced. Since the computational expense of the simulations is proportional to the number of simulated particles, it is necessary to minimize the number of particles while maintaining a sufficient number to preserve statistical accuracy. It can be concluded that a minimum of approximately 25 particles per sampling cell should be

maintained for accurate unsteady PDSC simulations, whilst maintaining acceptable computational times.

Figure 8 shows the effect of the cell size on the temperature profile at 27.45μs. Here in each case an average of 27.5 particles per cell were used. The variable *R* is the ratio of the cell size to the mean free path in the high-pressure gas. It can be seen that the cell size has a strong influence on the statistical scatter in the results, especially in the regions where the number of simulated particles is low. In this simulation accurate results can be maintained for quite a large cell size compared to the mean free path since the scale length of the macroscopic flow gradients are relatively large, however Bird has shown that in flows with large flow gradients the cell size should be kept to approximately one-third of the local mean free path [1]. This may become an issue within some simulations with large flow gradients.

As mention in section 2.3, one potential disadvantage of time averaging is that the flow structures will be "smeared" as the flow field develops over the sampling interval. For this reason, the sampling interval must be kept as short as possible to ensure smearing is minimised. To investigate the effect of smearing, the shock thickness was measured for Mach 4 and 8 moving shocks. Here a propagating one-dimensional shock was set up in PDSC from equations (1)-(4), with the initial conditions being a Maxwell-Boltzmann distribution on both sides of the shock. The shock was allowed to propagate downstream for 30λ for the Mach 4 case and 50λ for the Mach 8 case, to allow the true particle velocity distribution and shock structure to establish before the thickness was measured. Figure 9 shows a comparison of the shock thickness δ measured using the current method with the experimental results and other data from the paper by Schmidt [29]. The results are normalized with the mean free path upstream of the shock wave

(λ/δ). These results show the shock thicknesses obtained by unsteady sampling in PDSC are consistent with Schmidt's results. As long as the sampling period is constrained so that the majority of particles cannot migrate beyond the sampling cell during this time, smearing is minimised and accurate shock structures are maintained.

The results of the shock tube validation study confirm that the unsteady sampling procedures have been implemented correctly in PDSC. The results produced by PDSC are very similar to those produced by the established DS2V code and compare well with the continuum solution. Furthermore, the time-averaging sampling method gives comparable results to the ensemble-averaging method and PDSC using ten processors is approximately ten times faster than the single-processor DS2V code.

3.1.2. Validation of DREAM module

To test the DREAM-I scheme for improving the statistical scatter in the results, it was necessary investigate the validity of assuming a Maxwellian velocity distribution at every point in the flow when regenerating the particle data n sampling intervals prior to the output time for near-equilibrium (i.e. low shock Mach number) flows. To do this, the particle velocity distribution in the shock structure of the DREAM-processed data was compared to that in the raw data generated by PDSC.

Figure 10 shows the velocity distributions in the normal shock region at 76.9μs for the shock tube case given above. Both the distributions from the raw PDSC data and from those processed by DREAM-I are given, along with the curve for a Maxwellian distribution at the same temperature. Particle velocity v has been normalized by the macroscopic temperature T and macroscopic velocity v_m as $(v\text{-}v_m)(m/2kT)^{1/2}$. The parameter $N\Delta tW/\lambda_2$ represents the number of mean-free-paths which an average particle

will traverse in the N time steps of length Δt between regeneration and sampling, where W is the shock propagation velocity and λ_2 is the equilibrium mean free path immediately upstream of the shock.

The peak in the velocity distribution in the PDSC raw data is shifted to the left of the Maxwellian distribution, indicating the shock is sufficiently strong to be non-equilibrium. For the assumption of an initial Maxwellian distribution in DREAM-I to be valid, DREAM-I must reproduce the same profile as unsteady PDSC, rather than a Maxwellian distribution, thus indicating that the flow has relaxed rapidly enough towards an accurate distribution. When the particles have traveled less than one mean free path been regeneration and sampling, the distribution remains close to the Maxwellian. However, after approximately four mean free paths ($N\Delta tW/\lambda_2$=3.8), the particles have relaxed towards a similar distribution to that obtained by unsteady PDSC. This indicates that the assumption of a Maxwell-Boltzmann distribution in the regeneration step of DREAM-I is reasonably valid for Mach number flows below approximately Mach 2, providing the particles in the non-equilibrium regions are allowed to travel approximately four mean free paths between the regeneration and sampling time steps.

Figure 11 shows the temperature profile in the shock tube at 76.9μs as predicted by PDSC and after processing by DREAM-I with ten ensembled runs. Note temperature represents one of the macroscopic properties which has the highest statistical uncertainty in DSMC simulation. DREAM-I maintains the correct profile, while significantly reducing the statistical scatter in the results.

A quantitative measure in the reduction in statistical scatter obtained by DREAM-I can be determined by comparing the standard deviation of macroscopic properties in the

region of undisturbed flow. This represents the minimum reduction in statistical scatter, since the undisturbed flow is stationary and will therefore be the most scattered data in the simulation domain. Figure 12 shows the reduction in density scatter after DREAM processing with different numbers of ensemble runs and different starting points, compared to the original PDSC data. Here the number N of time steps Δt between regeneration and output are normalized with the velocity v_∞ and mean free path λ_∞ in the dense region of the flow. The scatter is normalized with the scatter in the unprocessed data.

Figure 12 allows the appropriate number of ensemble runs and regeneration data set to be determined. It shows that it is necessary to start sufficiently far from the output time step of interest to ensure good reduction in statistical scatter in the results. Regenerating the particle data at a time step too close to the output time results in a poor reduction in scatter, because the particles do not move far enough away from their regeneration positions, effectively constraining the final solution to be too close to the macroscopic properties of the regeneration data. Using a greater number of time steps and more ensemble runs results in a further reduction of scatter, however it is interesting to note that reduction in statistical scatter remains almost the same or only slightly different at larger $N\Delta tW/\lambda_2$ (=7.8) for both the ten and fifty ensembled runs. Thus, using larger number of ensembled runs with larger $N\Delta tW/\lambda_2$ is a case of diminishing returns and it should be noted that DREAM processing time is directly proportional to both the number of time steps and the number of ensembles in the sample.

Figure 13a shows the velocity distributions in a Mach 4 shock obtained using the assumption of a Maxwellian distribution (DREAM-I) in the regeneration data set. The data generated using DREAM-I (44,195 sampled particles) is compared to a separate

"high resolution" PDSC run with a large number of particles (62,824 sampled particles). Although the velocity distribution has relaxed somewhat by $N\Delta tW/\lambda_2 = 9$, there is still a considerable discrepancy in the peak of the distribution. For "engineering" type simulations, the use of the Maxwellian distribution assumption in higher Mach number flows may be justified, however when the correct particle velocity distribution profile is required in non-equilibrium regions, the use of particle data obtained from the original PDSC run is necessary. Figure 13b shows the velocity distributions obtained in the Mach 4 shock using this method (DREAM-II) for $N\Delta tW/\lambda_2 = 4.5$ (44,195 sampling particles), showing a much greater agreement with the velocity distribution from the high resolution PDSC run. The velocity distribution in the cell of interest at the sampling temporal point from the original PDSC run, which contains 2,076 sampling particles, is also shown. This run was used to generate the input phase-space data for DREAM-II and thus illustrates a major advantage of the method: initially PDSC can be run with a low number of simulated particles and DREAM-II can still obtain an accurate particle velocity profile, despite the scatter in the original data.

Figure 14 shows the particle velocity distribution obtained for a Mach 8 shock using DREAM-II which again demonstrates the ability of this method to obtain the correct particle velocity distribution in the shock. Here the high resolution PDSC data has 49,387 sampled particles and the DREAM-II result has 104,407 particles.

We have found a rule of thumb for selecting an appropriate regeneration data set is that the parameter $N\Delta tv_\infty/\lambda_\infty$ about four or greater where a Maxwellian is assumed for lower Mach number flows (i.e. Mach < 2) in DREAM-I. This allows the particle velocities to relax to the correct distribution while ensuring good reduction in the scatter of the macroscopic data while not incurring excessive computational expense. Where

the full phase-space data is used for higher Mach number flows in DREAM-II, $N\Delta t v_\infty/\lambda_\infty$ should also be set to about four to prevent constraining the macroscopic results to be close to the macroscopic data at the time of regeneration. We have also found that using approximately ten ensembles in the sample results in good reduction in statistical scatter while maintaining acceptable processing times.

3.2. Development of Couette Flow

Another method used to validate both the unsteady sampling techniques in PDSC and DREAM was the simulation of the development of Couette flow. The computational domain for this simulation is shown in figure 15. Here argon gas is initially at rest between two parallel diffuse plates at the same uniform temperature as the gas, in this case 300K. At time t=0 the upper plate begins moving instantaneously at speed U_∞=96.6 m/s. These conditions correspond to a Mach 0.3 flow with a Knudsen number of 0.02, based on the initial mean free path and the distance between the walls. Although this problem is one-dimensional, a 1m x 1m, 100 x 100 cell two-dimensional grid was used to help validate the code. This grid spacing was chosen to be half of the mean free path in the undisturbed gas. The simulation time step Δt was set at 3.11×10^{-5}s and TVTS was not used. ($\Delta t/t_c$ = 0.62, where t_c is the mean collision time of the stationary equilibrium gas).

A continuum solution for the velocity at the vertical position y and time t can be obtained from the incompressible Navier-Stokes equations [30]:

$$\frac{U(y,t)}{U_\infty} = \sum_{n=0}^{\infty} erfc\left[2n\eta_1 + \eta\right] + \sum_{n=0}^{\infty} erfc\left[2(n+1)\eta_1 - \eta\right] \qquad (5)$$

where $\eta = y/2\sqrt{\nu t}$, $\eta_1 = H/2\sqrt{\nu t}$, *erfc* is the complementary error function and *v* is the kinematic viscosity.

Figure 16 shows a comparison of the velocity profile from the raw PDSC data and the data after processing by DREAM-I as the flow reached steady state, illustrating the reduction in statistical scatter achieved by using DREAM. Without DREAM the level of velocity slip at the walls cannot be determined over the statistical scatter, however after processing the amount of slip is clearly discernable. Figure 17 shows the velocity profiles for a number of flow times as the Couette flow developed. All data has been processed by DREAM-I. In all cases time has been normalised such that $T = tU_\infty/H$.

Figure 17 shows that the PDSC/DREAM solution lags the incompressible continuum solution. This is because of compressible effects and because the high level of rarefaction effectively results in slip between gas particles and the walls. The PDSC/DREAM solution also exhibits the expected phenomenon of velocity slip at the walls.

4. Applications

4.1. Shock wave reflection over a wedge

The impingement of planar shock waves over wedges is a frequently studied problem for all levels of rarefaction. Experimental studies of relatively rarefied flows have been carried out by Walenta [31, 32] and comparable simulations using both DSMC and BGK solvers have been carried out by Xu *et al.* [22, 33]. At the continuum level, the impingement of a planar Mach 2 shock of ideal air over a 46º wedge is frequently used as a bench mark test for advanced numerical schemes in gas dynamics. A number of

experimental and simulation results from various researchers are given in the paper by Takayama and Jiang [34].

To demonstrate the capability of the proposed unsteady PDSC sampling procedures in successfully modelling these types of flow fields, a number of simulations of shock impingement over various wedge configurations were conducted. The test case was chosen to correspond to one of the experimental conditions of Walenta [32] and subsequently investigated using a BGK scheme by Xu and Honma [33]. Here a 25º wedge was simulated with a shock Mach number of 2.75 and VHS krypton as the molecular model. The Knudsen number based on width of the wedge normal to the flow and the high density flow to the left of the shock is 0.0019. The computational domain, shown in figure 18 consists of 77,899 unstructured sampling cells which were in turn divided into transient adaptive quadrilateral sub-cells which enabled nearest-neighbour collisions to be enforced. This sub-cell scheme will be described in more detail in a future publication. All domain boundaries were set as specular walls, except for the left hand inlet boundary which was set to the same conditions as behind the shock. A basic time step of $3\text{x}10^{-8}$s was used ($\Delta t/t_c = 0.0085$) with TVTS allowing the time step to increase by a factor of ten outside the sampling region. The number of particles in the domain peaked at approximately 7.2 million at the end of the simulation at which the shock reached a point $175\lambda_1$ from the leading edge of the wedge. The simulation time was 1.36 hours of simulation time on a 20-processor cluster similar to that described in section 3.1. A similar run without TVTS required 19.2 hours of simulation time. Post-processing of each data set using DREAM resulted in a reduction in the standard deviation of the density in the undisturbed region of flow from 20% in the original data to 9.6% in the processed data.

Figure 19 shows a comparison between the raw data from PDSC and the data processed by DREAM-I as the shock reaches approximately $175\lambda_1$ from the leading edge. Here the contours have been normalised by the densities in the undisturbed region ρ_1 and behind the shock ρ_2. In both cases, the reflected cylindrical shock, Mach stem and slip layer which form the triple point are clearly visible, however the structure is much better resolved in the DREAM processed data. In this figure, a density contour of $\frac{\rho-\rho_1}{\rho_2-\rho_1}=1.2$ from the equivalent case simulated by Xu and Honma [33] is also given, which shows good qualitative agreement with the present results.

Figure 20 shows the density contours at a further two points in the flow: when the shock reaches approximately $40\lambda_1$ and $110\lambda_1$ respectively. In both cases the data has been processed by DREAM-I. Again, in each case the resolution of the flow structure is greatly enhanced by using DREAM.

Figure 21 shows a comparison of the normalized density profile between the experimental data by Walenta [32], the BGK simulation by Xu and Honma [33] and the data from the present simulation at a point approximately $17.5\lambda_1$ behind the leading edge when the shock reaches $40\lambda_1$. Both the simulated profiles agree qualitatively, as they both use specular walls, however they differ from the experimental data due to the different wall conditions in the experiment. The PDSC/DREAM solution appears to exhibit higher density for $y/\lambda_1 < 10$ than the BGK solution.

4.2. Development of a shock wave structure passing a wedge in a channel

A further test problem used as a demonstration is the impingement of a planar Mach 1.3 shock over a two dimensional wedge in a channel. This problem was first studied in the classic experiment by Schardin who used high speed cinematography to study the

flow [35]. The flow results in a complex evolution of interacting shock and vortex structures. Numerous authors have also studied similar problems numerically using a Navier-Stokes solver by Huang [36], and using Euler equation solvers by Sivier *et al.* [37] and Chang and Chang [38], among others. The present simulation was developed as a point of comparison, however it should be noted that the present conditions are more rarefied than any of the cases mentioned above.

The computational domain for the problem is shown in figure 22. Here the channel wall and wedge surfaces were implemented as specular walls, and the inlet boundary was set to the same conditions as the behind the moving shock. Argon gas was used as the working gas, and the Knudsen number based on the width of wedge normal to the flow and the high density flow to the left of the shock was 0.012. For completeness the flow conditions are: p_1 = 10Pa, T_1 = 300K, p_2/p_1 = 1.8625, T_2/T_1 = 1.2922 and u_2 = 128.5 m/s, with these values being determined by the same method as in section 3.1.

The computational domain was divided into 147,011 unstructured elements, with the element size in the region of the wedge being approximately equal to the equilibrium mean free path of the conditions to the left of the shock. A sampling time step of $2\text{x}10^{-8}$s ($\Delta t/t_c$ = 0.049) was set such that no particle could traverse more than approximately one third of the mean free path over the sampling period, which consisted of 50 time steps. The TVTS scheme was employed, enabling the time step to be increased by a factor of ten when sampling was not occurring. The number of particles within the simulation domain peaked at approximately 21 million, and the total simulation required was approximately 3.5 hours of computational time on a 10-processor cluster with similar specifications as the cluster mentioned in section 3.1 (a run without TVTS required approximately 24 hours). The data sets at the points of

interest generated by PDSC were then post-processed using DREAM-I to improve the statistical accuracy of the simulations. In each case the solution was approached from 500 time steps before the point of interest and 10 ensembles were used. This procedure was found to reduce the standard deviation of the density in the undisturbed region of flow from 9.27% in the original data to 2.82% in the processed data and resulted in better resolution of the flow structure.

Figure 23 shows a series of density contours at different times as the shock passes over the wedge. The results reveal some very interesting flow field features, which are quantitatively consistent with the experimental and numerical results of the other authors mentioned above, although exhibit greater levels of rarefaction. As the incident shock passes over the wedge, the reflected cylindrical shock and Mach stem become clearly visible. At the end of the wedge, the Mach stem diffracts around the corner forming a further cylindrical shock and an expansion fan which moves in the opposite direction. The vortex structure formed behind the wedge then begins to move downstream. The cylindrical shocks from the upper and lower corner of the wedge cross each other (or, in the case of the simulation, reflect from the line of symmetry), which can be faintly seen in the figure. As the reflected shock from the front of the wedge grows larger, it reflects from the channel wall and then interacts with the shock structure below it.

5. Conclusions

In the current study, unsteady sampling methods for a parallel DSMC code were developed. To overcome the large computational expense and memory requirements usually involved in DSMC simulations of unsteady flows, several techniques were used.

Firstly, a method of time-averaging the flow near the sampling point was implemented which has considerable computational advantages over ensemble-averaging a large number of separate runs. Secondly, a temporal variable time step (TVTS) scheme was employed enabling faster processing between the sampling points without compromising simulation accuracy. TVTS resulted in substantial reductions in run-times due a larger time step outside the sampling zone resulting in reduced computational outlay on indexing and sampling the particles within the flow field. Thirdly, a method of post-processing certain sampling points, called DREAM, was developed whereby a combination of time- and ensemble-averaged data was build up by regenerating the particle data a short time prior to the sampling point of interesting, assuming either a Maxwell-Boltzmann distribution of particle velocities for Mach numbers less than about two or the original phase-space data obtained from the PDSC run for higher Mach number flows.

The validity of the unsteady sampling schemes was tested by simulating shock tube and unsteady Couette flow and a number of rules for appropriate use of the technique were developed. The code was then used to simulate a number of test cases involving the shock interaction over wedges. In each case, the technique produced accurate results comparable to other work in the literature with relatively low computational costs. In general it was found that ten ensembled runs of DREAM processing could reduce the statistical uncertainty in the raw PDSC data by 2.5-3.3 times, based on the limited number of test cases in the present study.

Acknowledgements

The authors gratefully acknowledge the assistance of W.-C. Hong in assisting with the simulations of unsteady Couette flow and of Y.-Y. Lian, T.-C. Kuo, M.-Z. Wu and M. Sellier for their valuable discussions on various aspects of this work. This work was partially financially supported by the New Zealand Tertiary Education Commission Top Achiever Doctoral Fund. The corresponding author J.-S. Wu was financially supported by the National Science Council of Taiwan through the project NSC-95-2221-E-009-018 and NSC-96-NU-7-009-001.

References

1. G.A. Bird, Molecular Gas Dynamics and the Direct Simulation of Gas Flows (Clarendon Press, Oxford, 1994).
2. J.-S. Wu and Y.-Y. Lian, Parallel three-dimensional direct simulation Monte Carlo method and its applications, Comput. Fluid 32(8) (2003) 1133.
3. I.D. Boyd, Y. Jafry and J.W. Beukel, Particle simulation of helium microthruster flows, J. Spacecraft Rockets 31 (1994) 271.
4. E.S. Piekos and K.S. Breuer, Numerical modeling of micromechanical devices using the direct simulation Monte Carlo method, J. Fluid Eng 118 (1996) 464.
5. H.A. Al-Mohssen and N.G. Hadjiconstantinou, Arbitrary-pressure chemical vapor deposition modeling using direct simulation Monte Carlo with nonlinear surface chemistry, J. Comput. Phys. 198(2) (2004) 617.
6. P. L. Bhatnagar, E. P. Gross, and M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one component systems, Phys. Rev. 94 (1954) 511.
7. J.Y. Yang and J.C. Huang, Rarefied flow computations using nonlinear model

Boltzmann equations, J. Comput. Phys. 120(2) (1995) 323.

8. W. Wagner, A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation, J. Stat. Phys. 66 (1992) 1011.
9. G.A. Bird, Sophisticated versus simple DSMC, in Proc. 25th Intern. Symp. on Rarefied Gas Dynamics, St. Petersburg, Russia, August 2006 (in press).
10. T.R. Furlani and J.A. Lordi, Implementation of the direct simulation Monte Carlo method for an exhaust plume flowfield in a parallel computing environment, AIAA Paper 88-2736 (1988).
11. Y. Matsumoto and T. Tokumasu, Parallel computing of diatomic molecular rarefied gas flows, Parallel Comput. 23 (1997) 1249.
12. R.P. Nance, R.G. Wilmoth, D. Moon, H.A. Hassan and J. Saltz, Parallel solution of three-dimensional flow over a finite flat plate, AIAA Paper 94-0219 (1994).
13. M. Ota and T. Tanaka, On speedup of parallel processing using domain decomposition technique for direct simulation Monte Carlo method. JSME (B) 57(540) (1991) 2696.
14. S. Dietrich and I. Boyd, Scalar and parallel optimized implementation of the direct simulation Monte Carlo method. J. Comput. Phys. 126 (1996) 328.
15. M. Ivanov, G. Markelov, S. Taylor and J. Watts, Parallel DSMC strategies for 3D computations, in Proc. Parallel CFD'96, Capri, Italy, May 1996, edited by P. Schiano, A. Ecer, J. Periaux and N. Satofuka (North-Holland, Amsterdam, 1997), p. 485.
16. G.J. LeBeau, A parallel implementation of the direct simulation Monte Carlo method. Comp. Meth. Appl. Mech. Engrs. 174 (1999) 319.
17. J.-S. Wu, K.-C. Tseng and F.Y. Wu, Parallel three-dimensional DSMC method

using mesh refinement and variable time-step scheme, Comput. Phys. Comm. 162 (2004) 166.

18. J.-S. Wu, K.-C. Tseng, Parallel DSMC method using dynamic domain decomposition, Int. J. Numer. Meth. Eng. 63 (2005) 37.
19. J.-S. Wu, S.-Y. Chou, U.-M. Lee, Y.-L. Shao, and Y.-Y. Lian, Parallel DSMC simulation of a single under-expanded free orifice jet from transition to near-continuum regime, J. Fluid Eng. 127 (2005) 1161.
20. H.M. Cave, S.P. Krumdieck and M.C. Jermy, Development of a model for high precursor conversion efficiency pulsed-pressure chemical vapor deposition (PP-CVD) processing, Chem. Eng. J.(2007) (in press).
21. D.J. Auld, Direct molecular simulation (DSMC) of shock tube flow, in Proc. First European Computational Fluid Dynamics Conference, Brussels, Belgium, September 1992.
22. D. Q. Xu, H. Honma and T. Abe, DSMC approach to nonstationary Mach reflection of strong incoming shock waves using a smoothing technique, Shock Waves 3(1) (1993), 67.
23. H.M. Cave, S.P. Krumdieck and M.C. Jermy, Simulations of reactor efficiency for pulsed pressure chemical vapor deposition (PP-CVD), in Proc. 25th Intern. Symp. on Rarefied Gas Dynamics, St. Petersburg, Russia, August 2006 (in press).
24. G.A. Bird, Monte Carlo simulation in an engineering context, Prog. Astronaut. Aeronaut. 74 (1981) 239.
25. K. Koura and H. Matsumoto, Variable soft sphere molecular model for inverse-power-law or Lennard-Jones potential, Phys. Fluids A 3 (1991), 2459.
26. J.-S. Wu, W.-J. Hsiao, Y.-Y. Lian and K.-C. Tseng, Assessment of conservative

weighting scheme in simulating chemical vapour deposition with trace species, Int. J. Numer. Meth. 43 (2003) 93.

27. K.-C. Tseng, J.-S. Wu and I. Boyd, Simulations of Re-Entry Vehicles by Using DSMC with Chemical-Reaction Module, AIAA-2006-8084, in Proc. 14th AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conf., Canberra, Australia, November 2006.
28. J.D. Anderson, Modern Compressible Flow: With Historical Perspective (McGraw-Hill, New York, 1990).
29. B. Schmidt, Electron beam density measurements in shock waves in argon, J. Fluid Mech., 39(2) (1969) 361
30. H. Schlichting and K. Gersten, Boundary Layer Theory (Springer, New York, 2000).
31. Z.A. Walenta, Microscopic structure of the Mach-type reflection of the shock wave, Arch. Mech. Warszawa 32(5) (1980) 819.
32. Z.A. Walenta, Formulation of the Mach type reflection of shock waves, Arch. Mech. Warszawa 35(2) (1983) 187.
33. D. Q. Xu and H. Honma, Numerical simulation for nonstationary Mach reflection of a shock wave: A kinetic-model approach, Shock Waves 1(1) (1991) 43.
34. K. Takayama and Z. Jiang, Shock wave reflection over wedges: a benchmark test for CFD and experiments, Shock Waves 7(4) (1997) 191.
35. H. Schardin, High frequency cinematography in the shock tube, J. Photographic Sci. 5 (1957) 19.
36. J.C. Huang, A study of instantaneous starting cylinder and shock impinging over wedge flow, in Proc. 10th National Computational Fluid Dynamics Conference,

Hua-Lien, Taiwan, August 2003 (in Chinese).

37. S. Sivier, E. Loth, J. Baum and R. Löhner, Vorticity produced by shock wave diffraction, Shock Waves 2(1) (1992), 31.

38. S.-M. Chang and K.-S. Chang, On the shock-vortex interaction in Schardin's problem, Shock Waves 10(5) (2000) 333.

List of Figures

Figure 10. Particle velocity distributions in the normal shock structure as produced by PDSC and DREAM-I with different values of $N\Delta tW/\lambda_2$

Figure 11. Comparison of shock tube temperature profiles at 76.9μs as predicted by PDSC and after processing by DREAM-I (ten ensembled samples).

Figure 12. Reduction in the statistical scatter of PDSC results following processing with DREAM-I for different numbers of ensembled runs.

Figure 13. Particle velocity distributions in the normal shock structure for a Mach 4 shock as produced by PDSC and a) assuming a Maxwellian distribution (DREAM-I) and b) using the original phase space data (DREAM-II).

Figure 14. Particle velocity distributions in the normal shock structure for a Mach 8 shock as produced by PDSC and using the original phase space data (DREAM-II).

Figure 15. Computational domain for the developing Couette flow verification case.

Figure 16. Comparison of raw PDSC data and data processed by DREAM-I as the Couette flow reaches steady state (T = 72)

Figure 17. Comparison of Couette flow development predicted by unsteady PDSC/DREAM-I (symbols) with the exact incompressible Navier-Stokes solution (lines). Note all times are normalized as $T = tU_\infty/H$.

Figure 18. Computational domain for shock impingement on a 25º wedge.

Figure 19. Comparison of (a) raw PDSC data and (b) data processed by DREAM-I for the impingement of a shock on a 25º wedge (Kn = 0.0019). The white markers show the 1.2 contour from the equivalent numerical simulation by Xu and Honma [33].

Figure 20. The impingement of a shock on a 25º wedge when the shock reaches approximately (a) $40\lambda_1$ (t = 280μs) and (b) $110\lambda_1$ (t = 600μs) from the leading edge of the wedge (Kn=0.0019). All data are processed by DREAM-I.

Figure 21. Comparison between the density distribution from experimental and simulation data at approximately 17.5λ_1 from the leading edge of the 25º wedge when the incident shock reaches 40λ_1.

Figure 22. Computational domain for the shock structure passing over a wedge in a channel.

Figure 23. Contours of density [kg/m^3] for shock impingement on a wedge in a channel (Kn=0.012). Each image is separated by 20μs

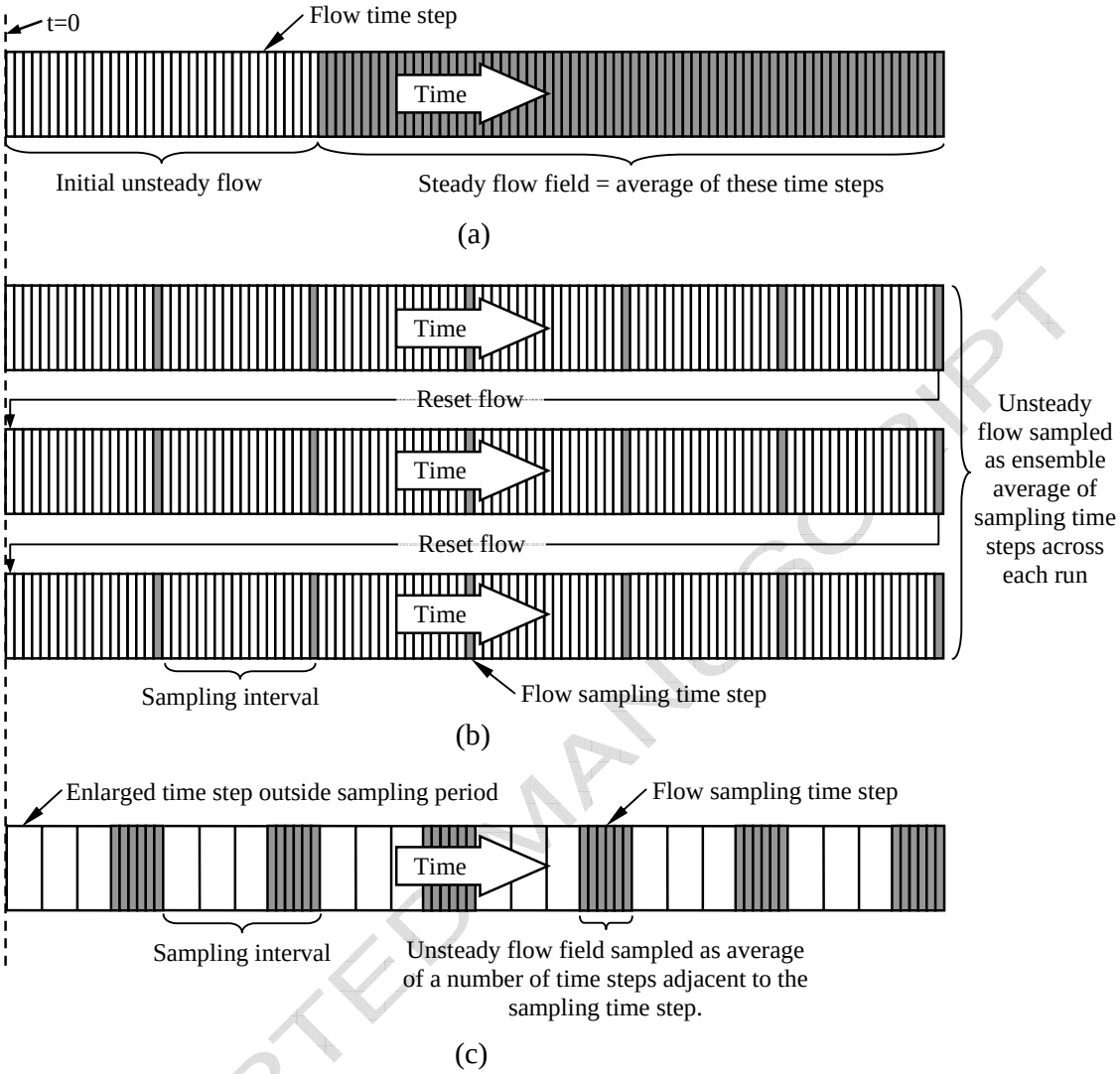

Figure 1. Sampling methods in DSMC including a) steady sampling, b) unsteady ensemble averaging and c) unsteady time averaging with temporal variable time step (TVTS).

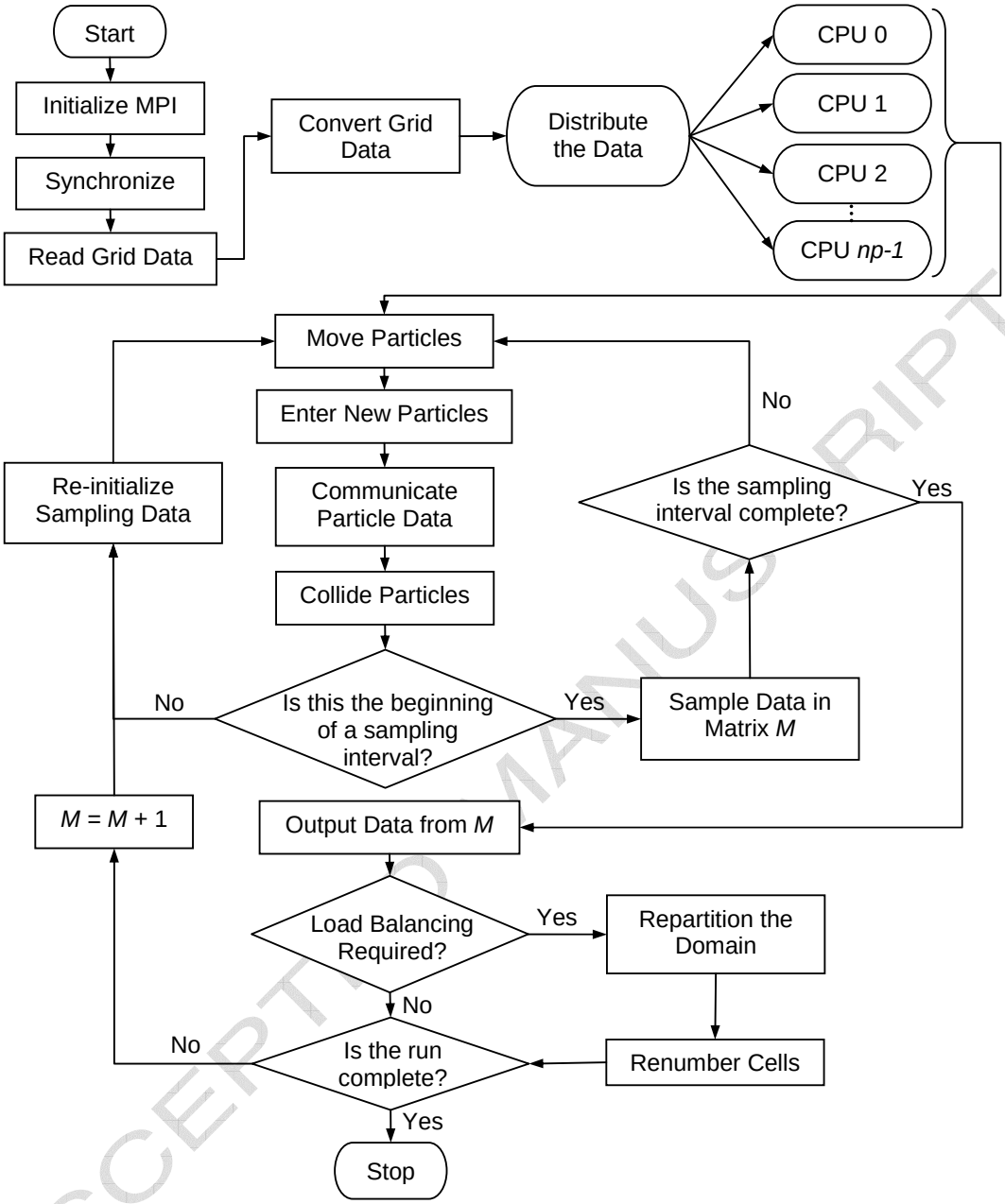

Figure 2. Simplified flow chart of the unsteady parallel DSMC method

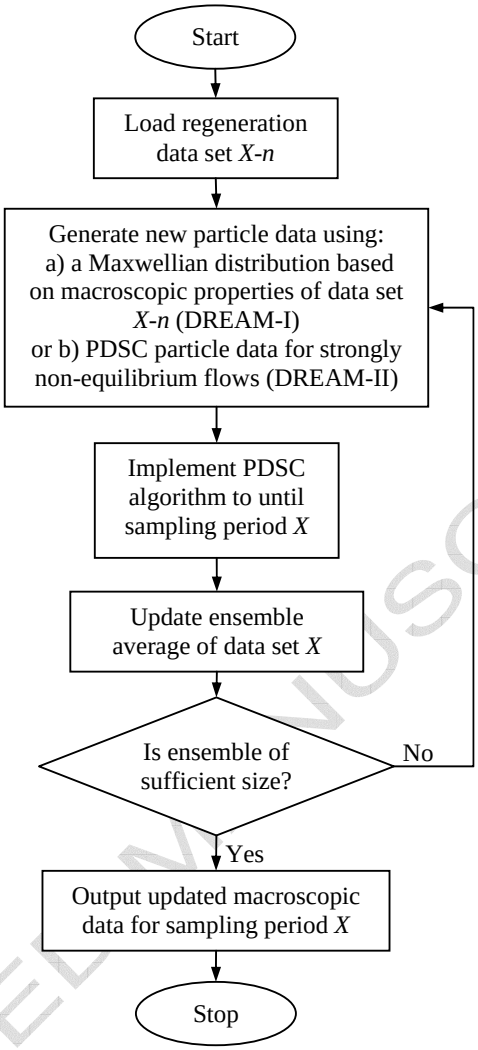

Figure 3. Simplified flow chart of the post-processing technique for unsteady DSMC sampling, called the DSMC Rapid Ensemble Averaging Method (DREAM)

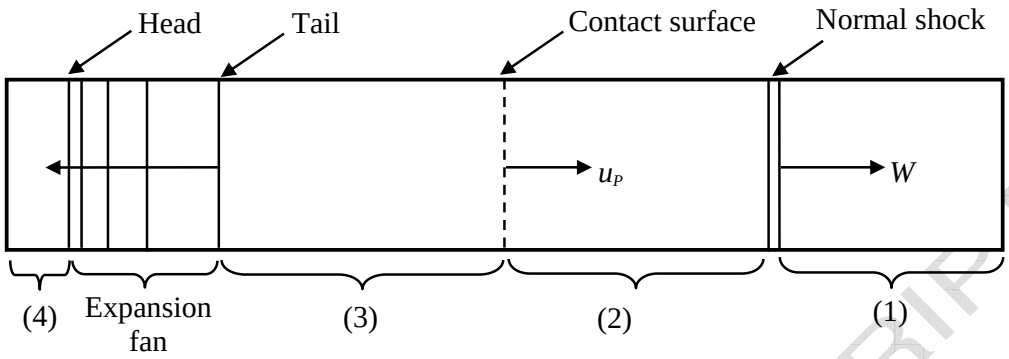

Figure 4. Flow structure in a shock tube. The flow regions include (1) the undisturbed low pressure gas, (2) the constant velocity gas behind the shock front, (3) the gas behind the contact surface between the driving and driven gases and (4) the undisturbed high pressure gas.

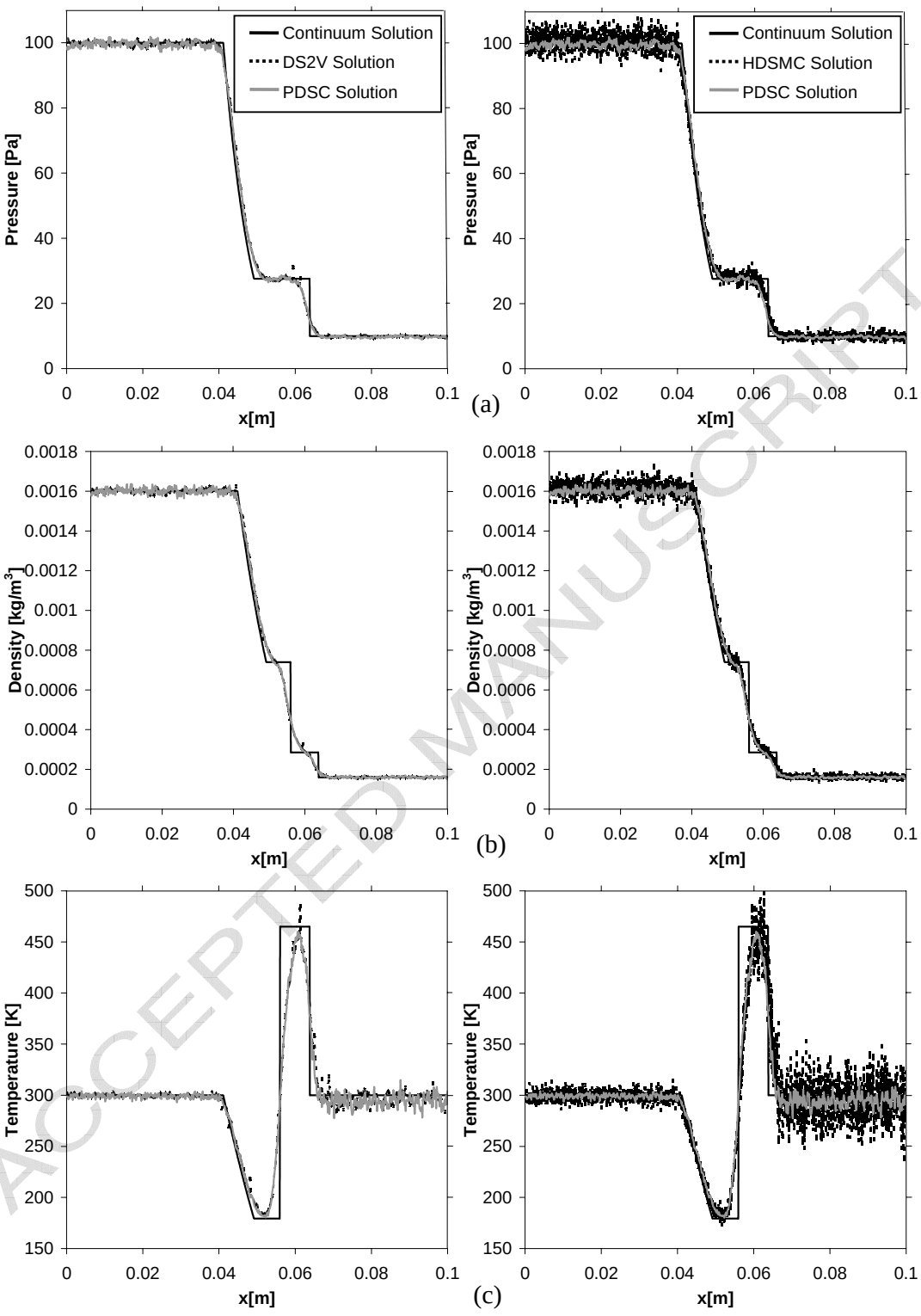

Figure 5. Shock tube flow field profiles of a) pressure, b) density and c) temperature at 27.45μs generated using PDSC (both sides), DS2V (left) and HDSMC (right).

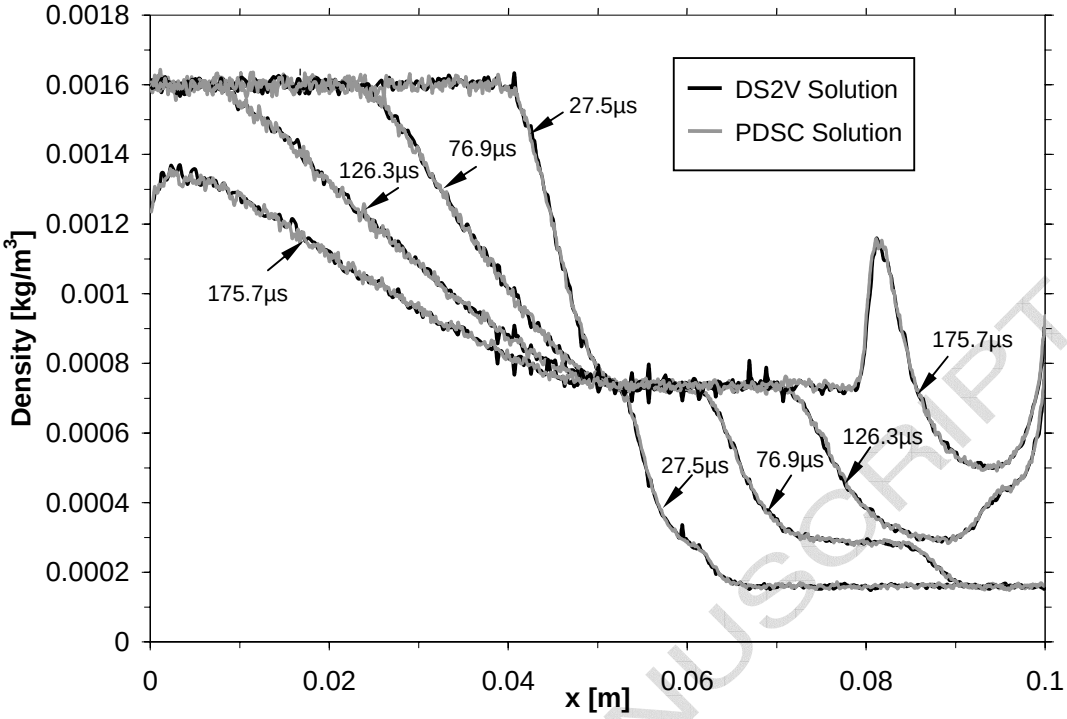

Figure 6. Evolution of flow structure in a shock tube as predicted by PDSC and DS2V

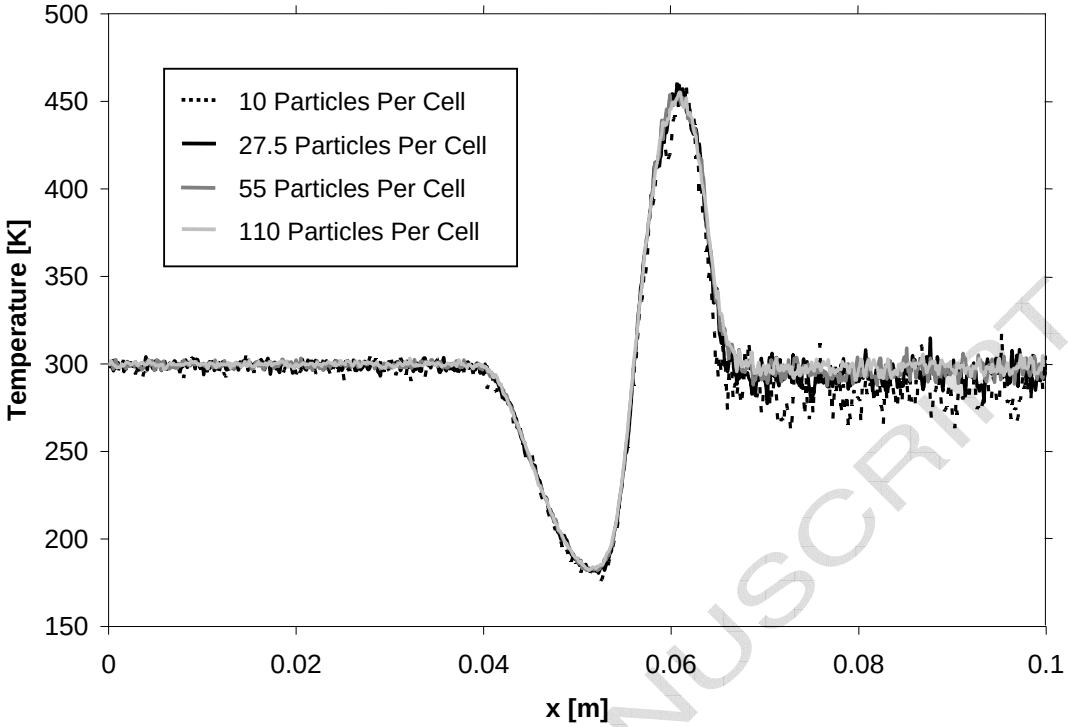

Figure 7. Effect of number of particles per cell on the shock tube temperature profile at 27.45μs (using 80,000 total cells).

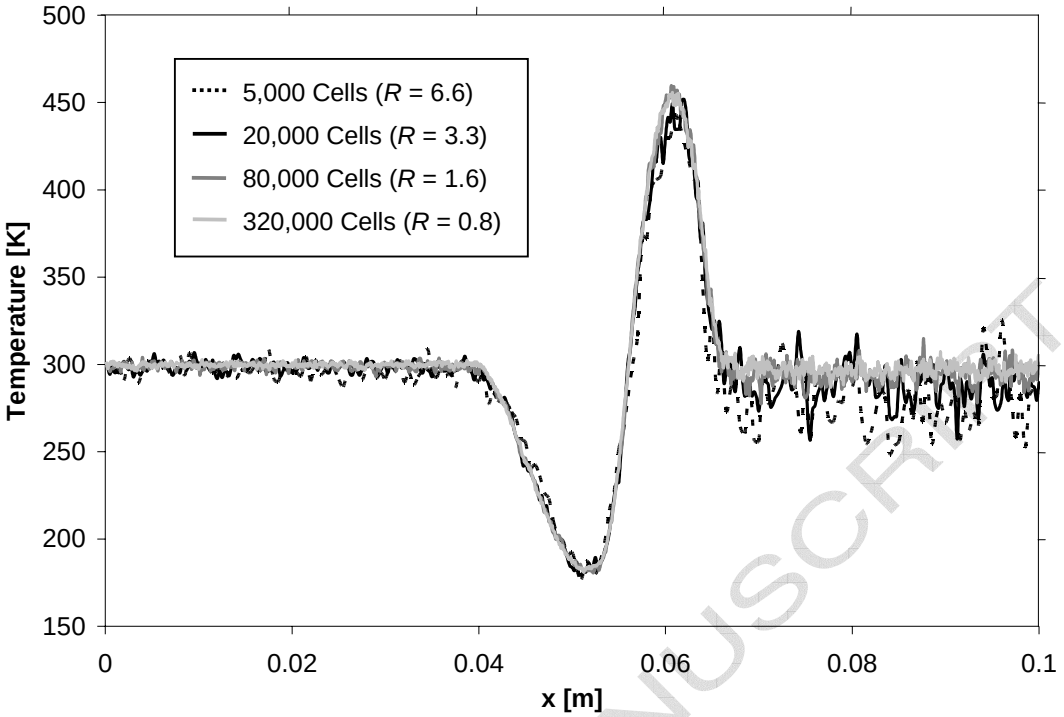

Figure 8. Effect of cell size with approximately 27.5 particles per cell on the shock tube temperature profile at 27.45μs.

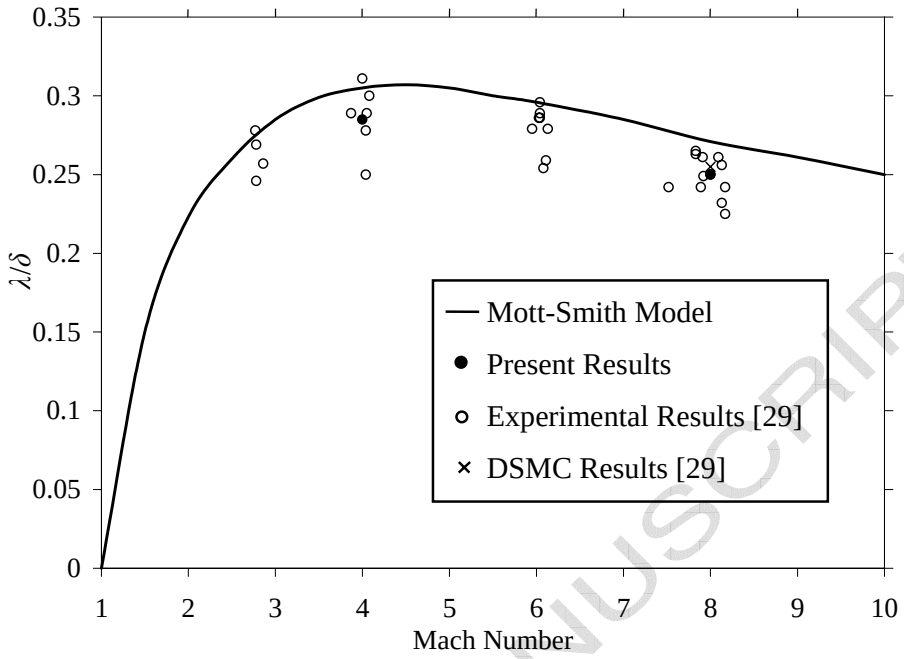

Figure 9. Normalized shock thickness predicted by PDSC compared to the data from reference [29].

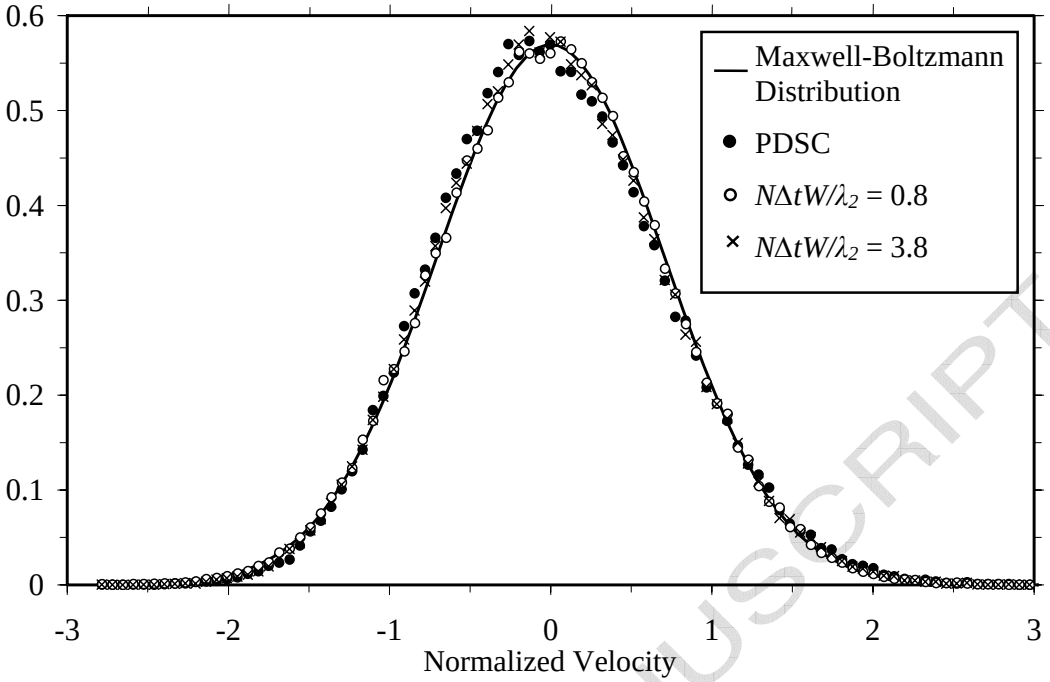

Figure 10. Particle velocity distributions in the normal shock structure as produced by PDSC and DREAM-I with different values of $N\Delta tW/\lambda_2$.

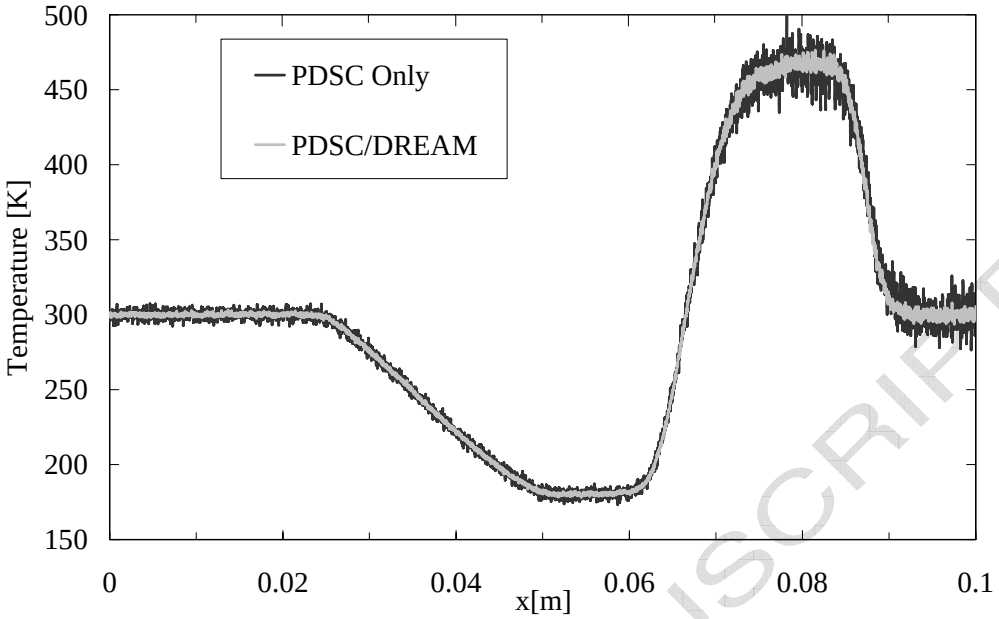

Figure 11. Comparison of shock tube temperature profiles at 76.9μs as predicted by PDSC and after processing by DREAM-I (ten ensembled samples).

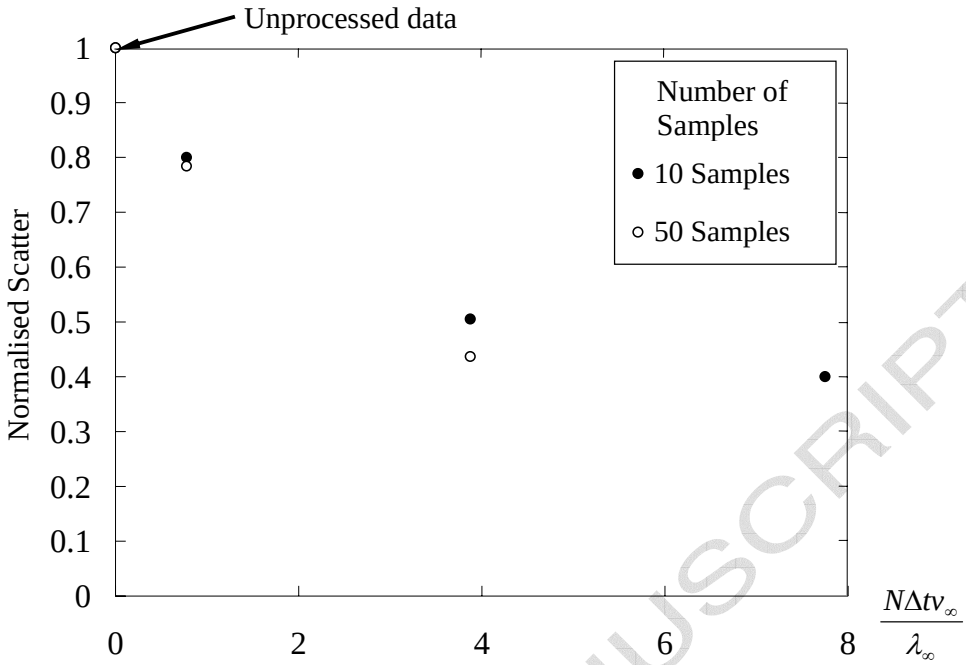

Figure 12. Reduction in the statistical scatter of PDSC results following processing with DREAM-I for different numbers of ensembled runs.

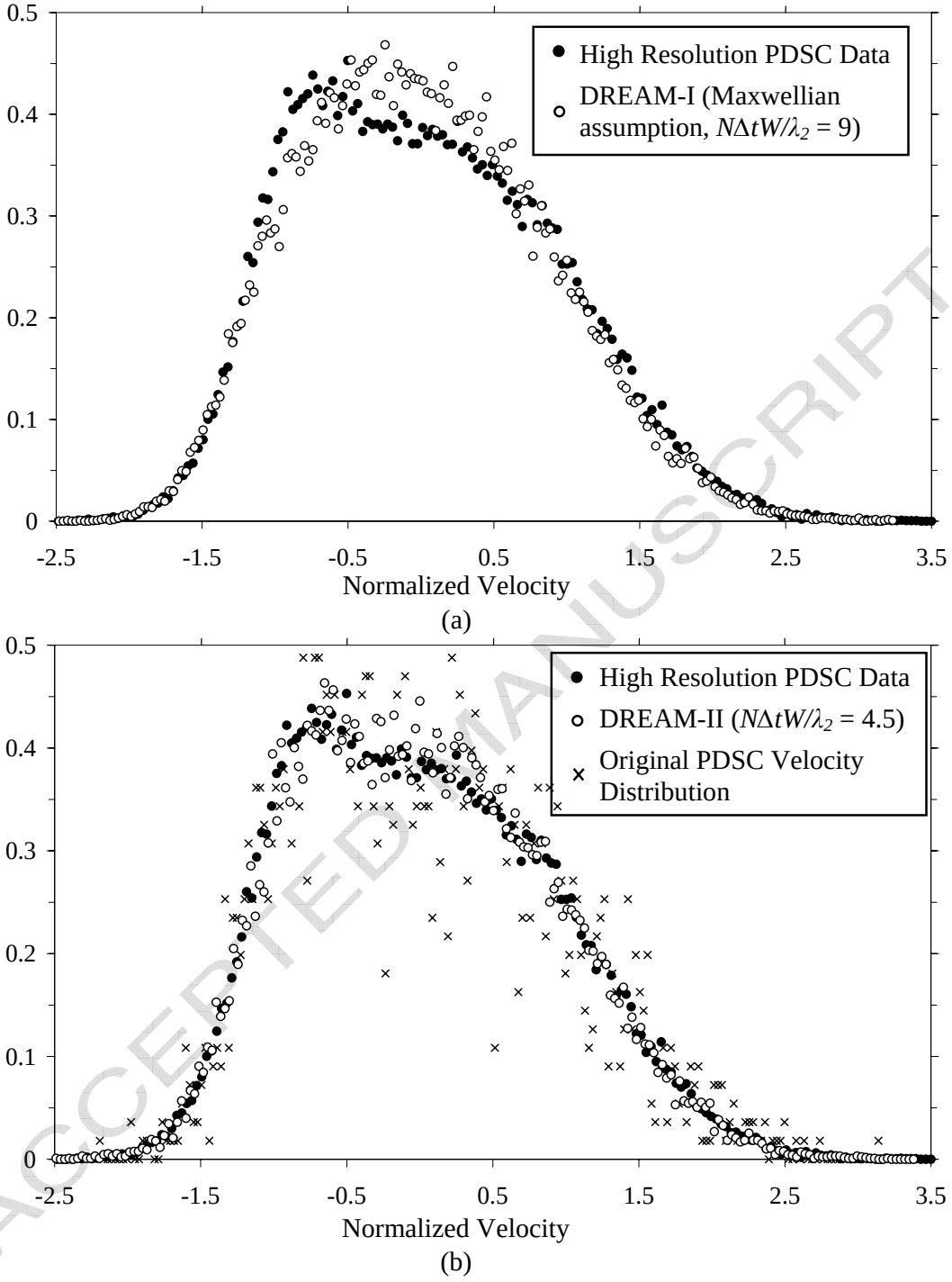

Figure 13. Particle velocity distributions in the normal shock structure for a Mach 4 shock as produced by PDSC and a) assuming a Maxwellian distribution (DREAM-I) and b) using the original phase space data (DREAM-II).

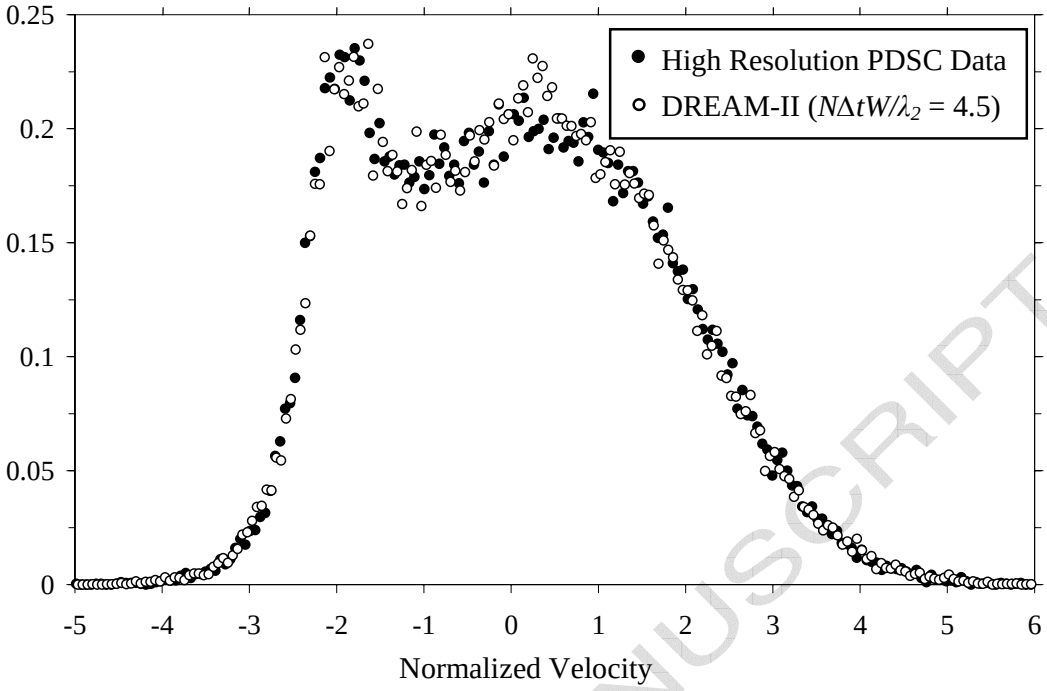

Figure 14. Particle velocity distributions in the normal shock structure for a Mach 8 shock as produced by PDSC and using the original phase space data (DREAM-II).

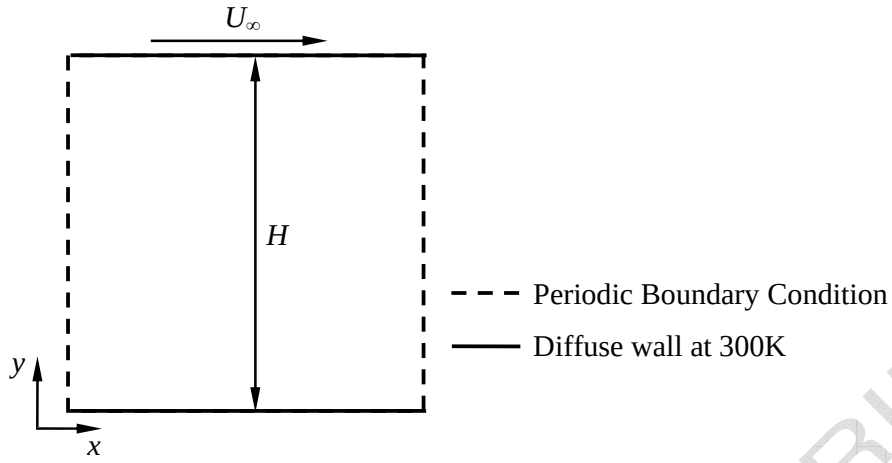

Figure 15. Computational domain for the developing Couette flow verification case.

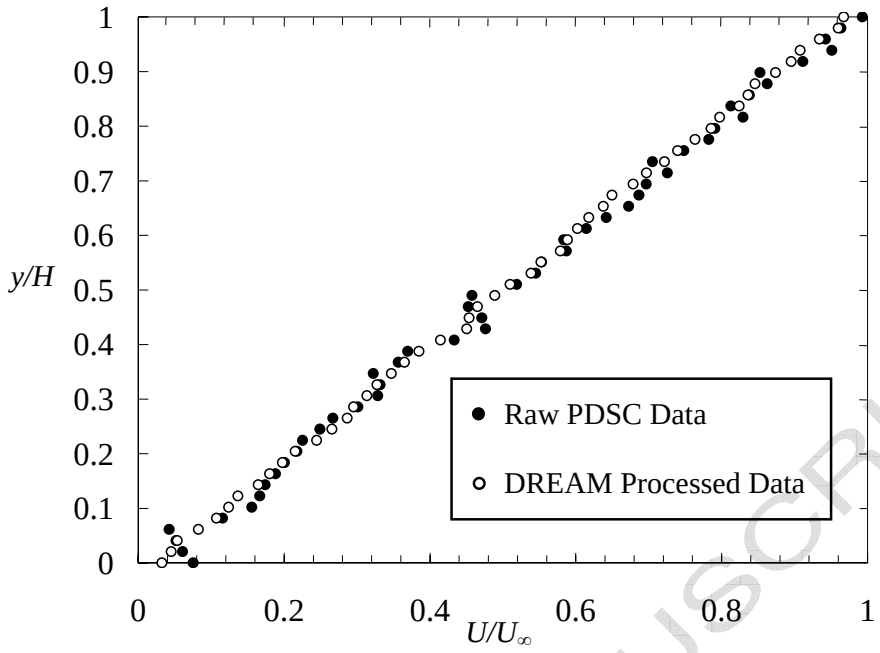

Figure 16. Comparison of raw PDSC data and data processed by DREAM-I as the Couette flow reaches steady state (T = 72)

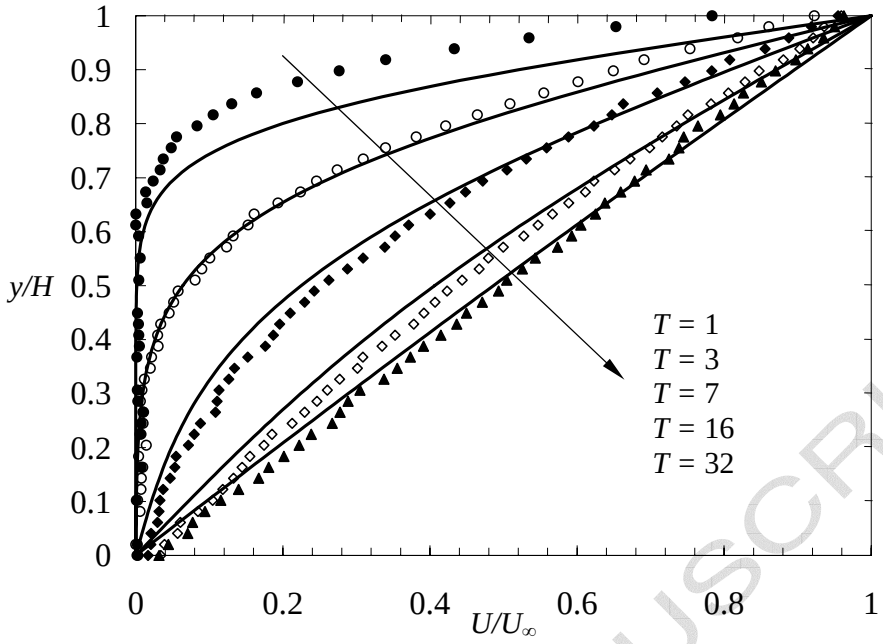

Figure 17. Comparison of Couette flow development predicted by unsteady PDSC/DREAM-I (symbols) with the exact incompressible Navier-Stokes solution (lines). Note all times are normalized as $T = tU_\infty/H$.

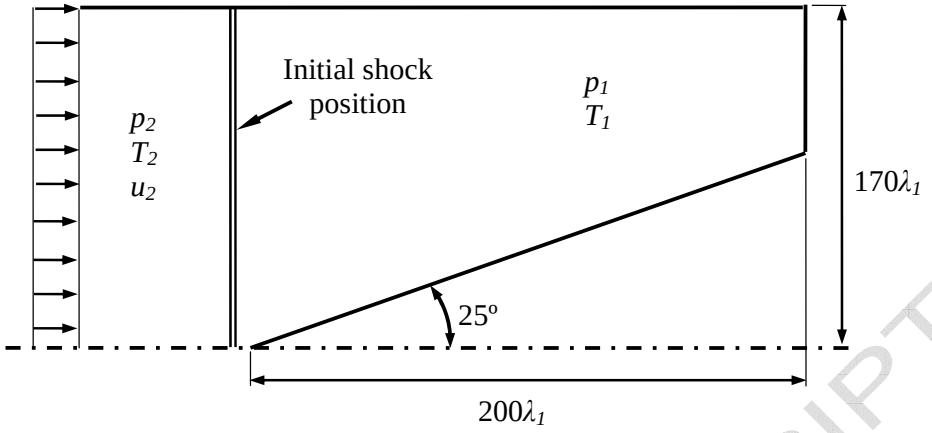

Figure 18. Computational domain for shock impingement on a 25º wedge.

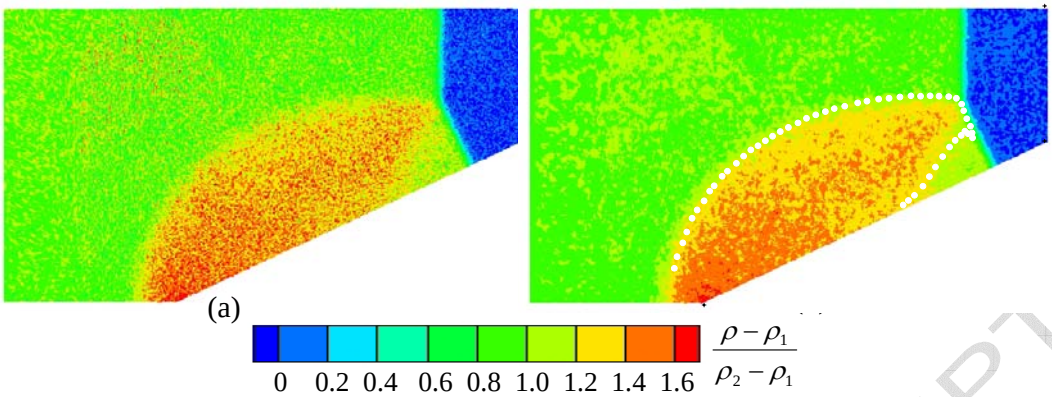

Figure 19. Comparison of (a) raw PDSC data and (b) data processed by DREAM-I for the impingement of a shock on a 25° wedge (Kn = 0.0019) at t = 900μs. The white markers show the $\frac{\rho-\rho_1}{\rho_2-\rho_1}=1.2$ contour from the equivalent numerical simulation by Xu and Honma [33].

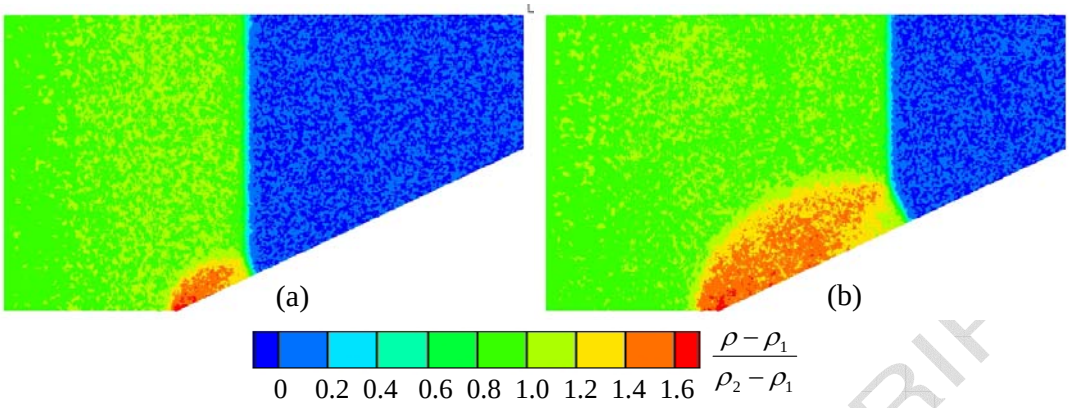

Figure 20. The impingement of a shock on a 25° wedge when the shock reaches approximately (a) $40\lambda_1$ (t = 280μs) and (b) $110\lambda_1$ (t = 600μs) from the leading edge of the wedge (Kn=0.0019). All data are processed by DREAM-I.

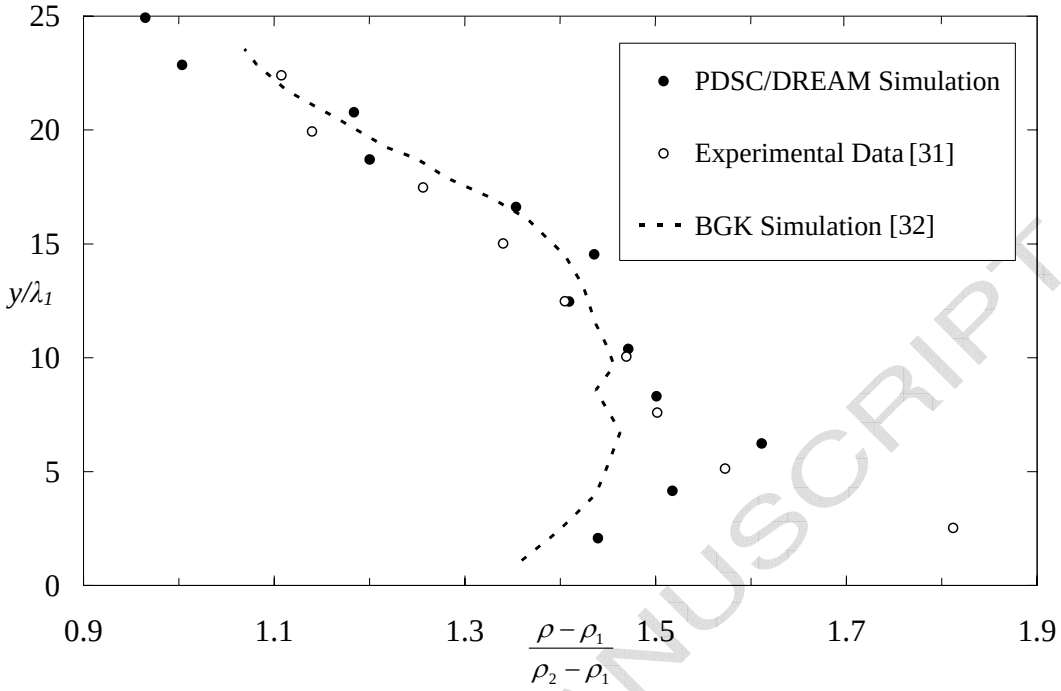

Figure 21. Comparison between the density distribution from experimental and simulation data at approximately $17.5\lambda_1$ from the leading edge of the 25º wedge when the incident shock reaches $40\lambda_1$.

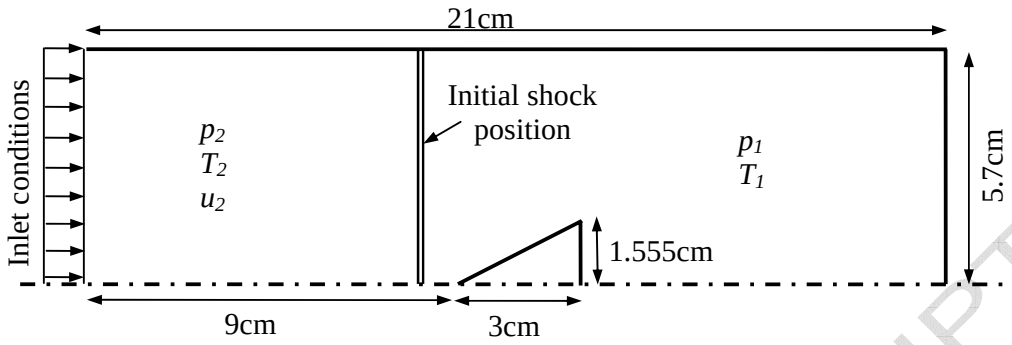

Figure 22. Computational domain for the shock structure passing over a wedge in a channel.

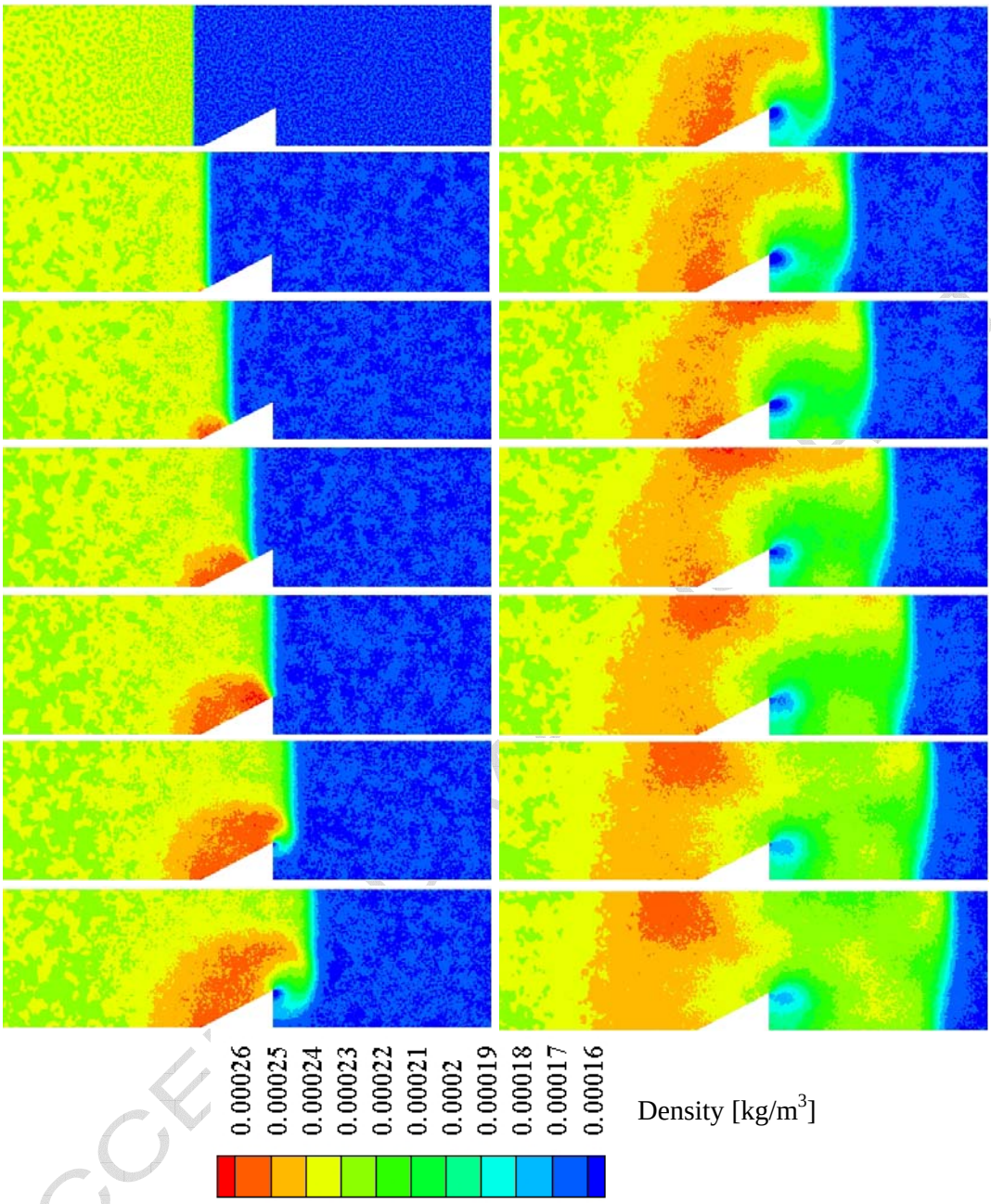

Figure 23. Contours of density [kg/m^3] for shock impingement on a wedge in a channel after processing by DREAM-I (Kn = 0.012). Each image is separated by 20μs

Table 1. Shock tube simulations using PDSC, DS2V and HDSMC

Solver	DS2V	PDSC	HDSMC
Number of Sampling Cells	21,675	80,000	2,000
Number of Collision Cells	~65,981	80,000	2,000
Number of Simulated Molecules	2,081,284	2,200,000	50,000
Time Step	Variable	9.15×10^{-8}s*	9.15×10^{-8}s*
Number of Time Steps Averaged Per Sample	30	30	N/A
Number of Simulations in the Ensemble Average	N/A	N/A	50
Run Time (hours per second of simulated flow)	~24,150 hrs/sec	~2,449 hrs/sec	~579,235 hrs/sec

*The basic time step in these runs was set to be equivalent to the basic time step employed by DS2V